1. thread.\*
2. thread.h
3. Thread class

新增int變數priority

新增int變數executionTime

新增int變數predictionTime

1. int getPriority()/void setPriority()
2. int getExecutionTime ()/void setExecutionTime()
3. int getPredictionTime ()/void setPredictionTime()/void calculatePredictionTime()
4. thread.cc
5. Thread constructor

priority = 149;

在Exec給予優先度

1. scheduler.\*
2. scheduler.h
3. 新增List<Thread \*> \*L1,L2,L3
4. scheduler.cc
5. void
6. Scheduler::ReadyToRun (Thread \*thread)

int priority = thread->getPriority();

int queueLevel = 0;

if(priority >= 100 && priority <= 149){

L1->Append(thread);

queueLevel = 1;

}

else if(priority >= 50 && priority <= 99){

L2->Append(thread);

queueLevel = 2;

}

else if(priority >= 0 && priority <= 49){

L3->Append(thread);

queueLevel = 3;

}

printf("Tick [%d]: Thread [%s] is inserted into queue L[%d]", kernel->stats->totalTicks, thread->getName(), queueLevel);

1. Kernel.\*
2. Kernel.h
3. Kernel
4. Exec增加int priority參數
5. 增加procees\_priority[10]=>用來吃輸入的priority，給予exec

Kernel.cc

1. Kernel::Kernel
2. 增加對”-ep”的判斷
3. ExecAll
4. 迴圈中的Exec裡新增參數
5. Exec
6. 新增new Thread後，要給與參數priority